

IBM Informix TOOLS Ver7.3 Migration Note

**日本アイ・ビー・エム株式会社
Customer Services**

Version Number:1.03 (日本語)

目次

- 1 . 紹介編 : P1 – P2
 - 1 . 1 はじめに
 - 1 . 2 IBM Informix Tool 7.3Xについて
- 2 . 新機能編 : P3 – P9
 - 2 . 1 DBCENTURY環境変数のサポート
 - 2 . 2 FET_BUF_SIZE環境変数のサポート
 - 2 . 3 PDQPRIORITY環境変数のサポート
 - 2 . 4 GLS環境変数のサポート
 - 2 . 5 アプリケーションレベルの新機能
- 3 . 機能変更編 : P10 – P15
 - 3 . 1 使用できる7.Xツール
 - 3 . 2 アプリケーション動作コードについて
 - 3 . 3 部分文字列の処理の変更
 - 3 . 4 スクリーンフォームとレポートの変更点
 - 3 . 5 提供されないライブラリ関数
 - 3 . 6 アプリケーションレベルの機能変更
- 4 . プロジェクト作業編 : P16 – P17
 - 4 . 1 マイグレーションのための参考資料
 - 4 . 2 TERMCAPファイルの確認及び修正
 - 4 . 3 インストール
 - 4 . 4 再コンパイル
 - 4 . 5 環境変数の確認

注意

1. 日本アイ・ピー・エム株式会社からの書面による承諾を得ずに、本書の内容の一部または全部を無断で、複製、複写、転載、翻訳、頒布することを禁止します。
2. 本書の内容は、製品の改良により将来予告なしに変更することがあります。
3. 本書の商品性、特定目的に対する適合性に関して、日本アイ・ピー・エム株式会社は保証いたしません。
4. 本書に関する御質問に対しては、弊社オープンラインにて本書に記述する範囲でお受け致します。
5. 本書の内容については万全を期して作成しましたが、万一記載内容の誤りなどお気づきの点がございましたら日本アイ・ピー・エム株式会社までご連絡ください。
6. 運用した結果の影響については、5 項にかかわらず日本アイ・ピー・エム株式会社は責任を負いかねますのでご了承ください。

1. 紹介編

1.1 はじめに

今回弊社では IBM Informix4GL及び IBM Informix-SQLVersion4/6からVersion7.3Xへアップグレードする際の技術的な作業に関連する内容を概説した資料を用意しました。本書はVersion4/6からVersion7.3Xへの移行についての課題に焦点を絞り、関連したトピックを説明するものです。

1. 紹介編

この章では、当資料の目的と IBM Informix4GL,SQL Ver7.3Xについての位置づけについて説明します。

2. 新機能編

この章では IBM Informix4GL,SQL Ver7.3Xの新しい機能についての概要を述べてあります。

3. 機能変更編

この章では IBM Informix4GL,SQL Ver7.3Xの特に重要と思われる変更点について述べてあります。

注意：新機能および機能変更編は共にその概要をわかりやすく伝える事を目的としています。従って、その他数多くの新機能や機能変更の詳細については下記のガイド類を参照してください。

『INFORMIX Migration Guide』	(日本語版あり)
『UNIX Products Migration Guide』	(日本語版あり)
『INFORMIX Installation Guide』	(日本語版あり)
『Guide to GLS Functionality』	(日本語版あり)

4. プロジェクト作業編

この章では実際のマイグレーションに必要なタスクをステップ順に記述しています。

注意：マイグレーションについては以下のガイドが用意されております。詳細についてはガイドをチェックしてください。またVersion7.30で追加・変更された機能については製品をインストールしたディレクトリにある、[release/en_us/0333/TOOLREL_7.30](#)にまた、Version7.31で追加・変更された機能については製品をインストールしたディレクトリにある、[release/en_us/0333/TOOLREL_7.31](#)に記述されています。必ず上記リリースノートは必ず参照してください。

『INFORMIX Migration Guide』	(日本語版あり)
『UNIX Products Migration Guide』	(日本語版あり)

1.2 IBM Informix4GL及びSQL Ver7.3Xについて

IBM Informix4GL バージョン7.3Xと Informix - SQL バージョン7.3X(以後7.3Xtoolと記述します)は UNIX環境においての、キャラクタベースのツールプロダクト群です。

7.3Xtoolは IBM Informix 4GL及び IBM Informix SQL Version4/6,7.2X製品のアップグレード版として位置づけられています。

7.3Xtoolプロダクトはそれぞれグローバル言語サポート(通称GLSと呼ばれています)を供給しており、日本を含むアジア、ヨーロッパ、ラテンアメリカ、そして中東の国のコードセットに対するサポートを提供しています。

2. 新機能編

7.3X toolは以前のバージョンより新しい機能をサポートしています。

2.1 DBCENTURY環境変数のサポート

環境変数DBCENTURYを利用することで、ユーザが日付型フィールドに年を表す2桁の数値のみを入力したときに、Informix製品がデフォルトで設定する世紀を制御することができます。また、環境変数DBCENTURYが設定されていなかったときには、コンピュータのシステムクロックから得られた世紀がデフォルトの世紀としてInformix製品内で利用されます。このデフォルトの動作は、DBCENTURYに対応せず20世紀の年をデフォルトにするこれまでのバージョンの動作とは異なっていることに注意してください。また、ユーザが4桁の数値を入力したときには、その数値がDBCENTURYの設定に関わらず年を表すものとして扱われます。

DBCENTURYには、ユーザが2桁の数値による年とともに日付を入力したときに適切な世紀を決定するために4種類のアルゴリズムを指定できます。また、実際の日付は、コンピュータのシステムクロックとの関係から決定されます。表4.1に、DBCENTURYに対する4種類の選択肢を示します。

設定値	意味	内容
R	現在	コンピュータのシステムクロックによる日付と同一の世紀が現在の世紀として利用されます。
P	過去	コンピュータのシステムクロックによる日付と比較して、それ以前のもっとも近い日付が利用されます。
F	未来	コンピュータのシステムクロックによる日付と比較して、それ以降のもっとも近い日付が利用されます。
C	最近	現在の日付にもっとも近い日付が利用されます。

表 4.1 DBCENTURY に対する 4 種類の設定値とその意味

以下に例を示します。

<例1. DBCENTURY=P - Past(過去)>

例 1.1

日付型例: date 型

現在日付: 4/6/1996

ユーザー入力日付 : 1/1/10

旧世紀への拡張: 1/1/1810

現行世紀への拡張 : 1/1/1910
分析 : 現在日付から見て上記の 2 つの日付はいずれも過去の判定となり、1/1/1910
のほうが現在に近い。
結果 : 1/1/1910

例 1.2

日付型例: date 型
現行日付 : 4/6/2010
ユーザー入力 : 1/1/05

旧世紀への拡張 : 1/1/1905
現行世紀への拡張 : 1/1/2005
分析 : 現行日付から見て上記の 2 つの日付はいずれも過去の判定となり、
1/1/2005 の方がより現在に近い。
結果 : 1/1/2005

例 1.3

日付型例 : date 型
現行日付 : 4/6/2010
ユーザー入力 : 1/1/50

旧世紀への拡張 : 1/1/1950
現行世紀への拡張 : 1/1/2050
分析 : 1/1/1950 のみが現行日付から見て過去の判定になる。
結果 : 1/1/1950

例 1.4

日付型例 : datetime year to month 型
現行日付 : 4/6/1996
ユーザー入力 : datetime(1-1) year to month
旧世紀への拡張 : 1801-1
現行世紀への拡張 : 1901-1
分析: 現行日付から見て上記の 2 つの日付はいずれも過去の判定になり、
1901-1 の方がより現在に近い
結果: 1901-1

<例 2. DBCENTURY=F - Future(未来)>

例 2.1

日付型例 : date 型
現行日付 : 4/6/1996
ユーザー入力 : 1/1/90

現行世紀への拡張 : 1/1/1990
次世紀への拡張 : 1/1/2090
分析 : 1/1/2090 が現行日付より未来になる
結果 : 1/1/2090

例 2.2

日付型例 : date 型
現行日付 : 4/6/2051
ユーザー入力 : 1/1/50

現行世紀への拡張 : 1/1/2050
次世紀への拡張 : 1/1/2150
分析 : 1/1/2150 が現行日付より未来になる
結果 : 1/1/2150

例 2.3

日付型例 : date 型
現行日付 : 4/6/2010
ユーザー入力 : 1/1/25

現行世紀への拡張 : 1/1/2025
次世紀への拡張 : 1/1/2125
分析 : 現行日付から見て、上記の 2 つの日付はいずれも未来の判定になり、1/1/2025
の方がより現在に近い。
結果 : 1/1/2025

例 2.4

日付型例 : datetime year to month 型
現行日付 : 4/6/1996
ユーザー入力 : datetime(1-1) year to month

現行世紀への拡張 : 1901-1
次世紀への拡張 : 2001-1
分析 : 2001-1 が現行日付より未来になる。
結果 : 2001-1

<例 3. DBCENTURY=C - Closest(最近)>

例 3.1

日付型例 : date 型
現行日付 : 4/6/1996
ユーザー入力 : 1/1/1

旧世紀への拡張 : 1/1/1801 から 4/6/1996 まで 71318 日
現行世紀への拡張 : 1/1/1901 から 4/6/1996 まで 34794 日
次世紀への拡張 : 1/1/2001 から 4/6/1996 まで 1731 日
分析 : 1/1/2001 が現行日付にもっとも近い
結果 : 1/1/2001

例 3.2

日付型例 : date 型
現行日付 : 2/1/2001
ユーザー入力 : 4/1/95

旧世紀への拡張 : 4/1/1995 から 2/1/2001 まで 2133 日
現行世紀への拡張 : 4/1/2095 から 2/1/2001 まで 34392 日
次世紀への拡張 : 4/1/2195 から 2/1/2001 まで 70916 日
分析 : 4/1/1995 が現行日付に最も近い。
結果 : 4/1/1995

例 3.3

日付型例 : datetime year to month 型
現行日付 : 4/6/1996
ユーザー入力 : datetime(1-1) year to month

現行日付 までの月数が計算される。
旧世紀への拡張 : 1801-1 から 1996-4 まで 2343 ヶ月
現行世紀への拡張 : 1901-1 から 1996-4 まで 1143 ヶ月
次世紀への拡張 : 2001-1 から 1996-4 まで 57 ヶ月
分析 : 2001-1 が現行日付に最も近い。
結果 : 2001-1

<例 4. DBCENTURY=R - Present(現在)>

例 4.1

日付型例 : date 型
現行日付 : 4/6/1996
ユーザー入力 : 12/10/1

結果 : 12/10/1901

例 4.2

日付型例 : date 型
現行日付 : 4/6/2010
ユーザー入力 : 12/10/1

結果 : 12/10/2001

例 4.3

日付型例 : datetime year to month 型
現行日付 : 4/6/1996
ユーザー入力 : datetime(1-12) year to month

結果 : 1901-12

2.2 FET_BUF_SIZE環境変数のサポート

fetch bufferはデータベースサーバーがBLOBデータ以外のデータをアプリケーションに送るために使用するバッファです。このバッファはアプリケーションプロセス内に常駐しています。バッファを大きくすると、アプリケーションに制御が戻る前に、データベースサーバーからアプリケーションへより多くのデータを送ることができます。

非常に多くの行を返すようにするために、バッファサイズを大きくすることにより、データベースサーバーが行を抽出する間にアプリケーションが待機しなければならない回数が少なくて済むようになります。

バッファサイズの変更を行うには環境変数FET_BUF_SIZEを設定します。

* 以下のコマンドによって C-shell 環境下で環境変数FET_BUF_SIZEが20,000バイトに設定されます。

```
setenv FET_BUF_SIZE 20000
```

指定するサイズはバイト単位で、使用しているシステムの小桁整数(short型)の最大値が上限になります。ほとんどの32ビットシステムではこの最大値は32,767byteです。
バッファサイズを指定しない場合はデフォルトのサイズが使用されます。

注：fetch buffer(取り出しバッファ)についての詳細は『INFORMIX-ESQL/C Programmer's Manual』を参照してください。

2.3 PDQPRIORITY環境変数のサポート

OnLine サーバーがパラレルクエリプロセスに割り当てるリソースを決める環境変数がPDQPRIORITY環境変数です。

設定は以下のように行います。

(cshell環境)

```
setenv PDQPRIORITY-----HIGH
      |-----LOW
      |-----OFF
      |-----percent-of-resources
```

Element	Description
HIGH	OnLineがすべてのユーザーに資源を割り当てる場合、問い合わせに対して可能な限りの資源を与えます。
LOW (Default)	フラグメントされた表からデータの取り出しは並列して行われますが、それ以外の並列処理は行われません。
OFF	PDQ処理はオフになります
Percentage-of-resources	問い合わせの優先レベルを示す0から100の整数値です。この数字が大きくなればなるほど、OnLineが使用する資源の量は多くなります。 0はoffを指定したのと同じです。 1はlowを指定したのと同じです。

2.5 GLS環境変数のサポート

7.2Xtool及び7.3Xtool製品はGLS環境変数のサポートを提供しています。

以下のような環境変数が含まれます。

* GL_DATE
* GL_DATETIME

GLS環境変数の詳細については『Guide to GLS Functionality』を参照してください。

2.6 アプリケーションレベルの新機能

これらの機能についての詳細はリリースノート及び『Version7.3 4GL Reference Manuals』を必ず参照してください。

2.6.1 連結演算子(“||”)の追加

ANSIのルールに従って連結演算子 (“||”)が使用可能になりました。
例)

```
DEFINE a CHAR (30)
LET a = "Informix " || "Development " || "Center"
DISPLAY "a=",a,"**"
```

結果 → [a=Informix Development Center*]

2.6.2 スクリーン配列での新しい設定

INPUT ARRAYアトリビュートの INSERT/DELETE keyについて以下のような設定が新しく追加されました。

```
INSERT ROW          -- signifies that INSERT KEY is enabled
INSERT ROW = TRUE   -- signifies that INSERT KEY is enabled
INSERT ROW = FALSE  -- signifies that INSERT KEY is disabled

DELETE ROW          -- signifies that DELETE KEY is enabled
DELETE ROW = TRUE   -- signifies that DELETE KEY is enabled
DELETE ROW = FALSE  -- signifies that DELETE KEY is disabled
```

2.6.3 レポート出力先の動的指定

実行時にレポートの出力先の指定が動的に可能になりました。
以下に例を示します。

例)

```
START REPORT reportname
  TO OUTPUT "pipe" DESTINATION "Ip -dcypress"
  WITH PAGE LENGTH = 64, TOP MARGIN = 0,
  BOTTOM MARGIN = 0, LEFT MARGIN = 0
```

2.6.4 CURRENT ROW DISPLAY アトリビュートの追加

INPUT ARRAY文及びDISPLAY ARRAY文に対して入力及び出力操作中にカレント行がハイライトされる CURRENT ROW DISPLAYアトリビュートが追加されました。

2.6.5 新しい関数

以下の新しい関数がVersion7.3X Informix4GLに追加されました。

FGL_GETKEY
FGL_SCR_SIZE
FGL_SETCURRLINE

これらの関数の詳細は『Version 7.3 4GL References Manuals』を参照してください。

3. 機能変更編

3.1 使用できる7.Xツール

IBM Informix-OnLine及び IBM Informix Dynamic Server製品(以後エンジン製品と呼びます)と7.3Xtoolの各バージョンでの接続および機能の制限を示します。

	エンジン製品		
	OnLine 5.1	OnLine 5.2	Dynamic Server
接続	ネットワーク接続のみ	ネットワーク接続可 ストリームパイプ接続可	ネットワーク接続可 ストリームパイプ接続可 共有メモリ接続可
SQL文の制限	Dynamic Server 製品で新しく追加された構文を使用することはできません。但し、prepare文を使用することによりInformix4GLで使用することは可能です。		なし

3.2 アプリケーション動作コードについて

アプリケーションプログラム内部で保持する文字列データのコードをアプリケーション動作コードと呼びます。バージョン4.XのTool製品で作成したアプリケーションプログラムとバージョン6.0X以降のTool製品で作成したアプリケーションプログラムでは、アプリケーション動作コードが異なる場合があります。

バージョン4.Xのアプリケーションプログラムでは、アプリケーション動作コードは内部コードと同じコードセットになるように設定されていました。6.0X以降のバージョンのアプリケーションプログラムでは、アプリケーション動作コードはAPI(アプリケーションプログラムインターフェイス)コードと同じコードセットになるように設定されています。このため、内部コードとAPIコードに異なるコードセットが指定されていると、バージョン4.Xのアプリケーションプログラムとバージョン6.0X以降のアプリケーションプログラムでは、アプリケーション動作コードが異なります。

アプリケーション動作コードが異なると、データベースのスキーマやアプリケーションプログラムの変更が必要になる場合があります。これは、sjis-sとujisでは、JIS X 0201片仮名符号やシフトJISユーザ定義の1文字あたりに必要となるバイト数が次のように異なるためです。

文字の種類	sjis-s	ujis
JIS X 0201片仮名符号	1バイト	2バイト
シフトJISユーザ定義	2バイト	3バイト

たとえば、バージョン4.Xのアプリケーションプログラムでは、内部コードがシフトJISに固定されていたため、アプリケーション動作コードもつねにsjis-sとなり、内部に10バイトの文字列格納領域があればJIS X 0201片仮名符号が10文字分格納できました。

バージョン6.0X以降のアプリケーションプログラムでは、APIコードにujisが指定されていると、アプリケーション動作コードもujisとなるため、10バイトの文字列格納領域には5文字のJIS X 0201片仮名符号しか格納できません。

10文字のJIS X 0201片仮名符号を格納するには、文字列格納領域を20バイトに変更しなければなりません。

また、スキーマを変更した場合、さらにスキーマに依存するアプリケーションプログラム内の記述も変更が必要となります。

このように、バージョン間でアプリケーション動作コードが異なる場合には、注意が必要です。

3.3 部分文字列の処理の変更

バージョン7.3の IBM INFORMIX製品では、部分文字列の処理が変更されています。文字 (CHAR) 型や可変長文字 (VARCHAR) 型で収められているデータに対して、添字を使用することで部分文字列を得ることができます。

ただし、添字には文字列のバイト単位での位置しか記述できないため、複数バイトによって構成されている日本語文字の2バイト目を指定した場合などに、意味のない単一バイトの文字が部分文字

列内に含まれることがあります。このような文字を欠損文字と呼びます。

通常、バージョン4.X/5.0Xの IBM INFORMIX製品では、欠損文字が生じてても何も処理は行われません

でした。バージョン7.3の IBM INFORMIX製品では、欠損文字を単一バイトの空白文字に置き換えます。このため、複数バイトによって構成されている日本語文字を格納した文字型や可変長文字型のデータに対して添字を使用して問合せを行った場合、バージョン4.X/5.0Xの IBM INFORMIX製品とバージョン7.3の IBM INFORMIX製品では得られる文字列の結果が異なります。

したがって、文字列のどの位置に複数バイト文字が存在しているかを考慮して、欠損文字を生じさせないようにしてください。また、欠損文字が生じた場合にバージョン7.3の IBM INFORMIX製品の処理に対応するように、バージョン4.X/5.0Xの IBM INFORMIX製品で作成したアプリケーション

プログラムを変更してください。

3.4 スクリーンフォームとレポートの変更点

バージョン4.Xのツール製品群とバージョン7.3Xtoolでは、アプリケーション動作コードの設定が異なります。この違いに伴ってアプリケーションプログラムの変更が必要になる場合があります。詳細については、3.2「アプリケーション動作コード」を参照してください。

スクリーンフォームとレポートに関する次の機能が変更されています。

- * フォーマット文字列
- * PICTURE属性
- * キーワードOPTIONS NODELIMITERS
- * WORDWRAP属性
- * ヤंक/プット機能
- * 編集キーの再割当て機能

3.4.1 フォーマット文字列

USING演算子で指定する日付型と数値型のデータの書式に使用するフォーマット文字列が変更されています。

日付型データの書式に使用するフォーマット文字列はバージョン4.1Xのツール製品では、フォーマット文字列の元号を表す文字として“n”を指定していました。7.3Xtool製品では、元号を表す文字として“g”または“e”を指定します。次に、元号を使用した書式例を示します。

tool バージョン		
元号付き日付	バージョン4.1X	バージョン7.2X/7.3X
H07/01/24	“nyy/mm/dd”	“gyy/mm/dd”
平成07年01月24日	“nnnnyy年mm月dd日”	“eyy年mm月dd日”

データを入力する際には“g”又は“e”を先頭に入力し年月日のデータを入力してください。以下に入力例を示します。

フォーマット指定	入 力	表示結果
gyy/mm/dd	“ H10/01/23 ”	H10/01/23
eyy/mm/dd	“ H10/01/23 ”	平成10/01/23

注意:バージョン4.3Xツールの日付型フォーマット文字列の文字指定及び入力は、7.3toolと同様の変更が行われています。

数値型データの書式に使用するフォーマット文字列のうち、通貨記号として単一バイトの円記号(¥)を使用できません。

通貨記号として使用できるフォーマット文字列は、ドル記号(\$)のみです。ただし、環境変数DBMONEYに円記号を指定することで、円記号を表示できます。次に、バージョン4.Xツールと7.3Xtoolの出力結果を示します。

	バージョン4.X	7.2Xtool/7.3Xtool
フォーマット文字列	“¥¥,¥¥¥”	“\$\$,\$\$\$”
数値	1234	1234
DBMONEYの設定	¥.	¥.
出力結果	¥1,234	¥1,234

注意: その他の変更項目についての詳細は『INFORMIX Migration Guide』内の「INFORMIX-SQLの変更点」及び「INFORMIX4GLの変更点」を参照してください。

3.5 提供されないライブラリ関数

バージョン6.0X以降のINFORMIX4GLで提供されないライブラリ関数は、次の5つに大別することができます。

- ・日本語文字処理のためのライブラリ関数
- ・変換表を使用する文字列変換のためのライブラリ関数
- ・漢数字変換を行うライブラリ関数
- ・日本語文字処理のためのC言語のライブラリ関数
- ・環境変数の設定値を得るためのライブラリ関数

7.3XのInformix4GLでは、次のライブラリ関数は提供されません。

(日本語処理のためのライブラリ関数)

- kj_atoj
- kj_ctype
- kj_jtoa
- kj_ljus
- kj_moji
- kj_rjust
- kj_width

(変換表を使用する文字列変換のためのライブラリ関数)

- kj_map

(漢数字変換を行うライブラリ関数)

- kj_nun
- kj_date

(日本語文字処理のためのC言語のライブラリ関数)

- kccconv
- kcnconv
- kclen
- kcnlen

(環境変数の設定値を得るためのライブラリ関数)

- getienv

バージョン7.3XのInformix4GLでは、これらのライブラリ関数と同等の機能を持つライブラリ関数は提供されません。ただし、使用しているUNIXシステムによっては、これらのライブラリ関数の代わりに使用できる関数が用意されている場合があります。

3.6 アプリケーションレベルの機能変更

以下の変更は7.3Xtoolのみの機能変更となります。

3.6.1 グローバル文字領域サイズの変更

INFORMIX4GL RDS で使用できるグローバル文字領域のサイズが従来の 64Kbytesから 2Gbytesにまで拡張されました。

3.6.2 拡張 SQL 構文が使用可能

以前の 4GL コンパイラーでは認識できなかった拡張SQL構文を 'SQL' ... 'END SQL' の形式の中に記述することにより拡張 SQL 構文をコンパイルできるよう4GL言語が拡張されました。

以下に例を示します。

例 1) パラメータが無いSQL文

```
SQL
CREATE SCHEMA AUTHORIZATION "andy"
CREATE TABLE T1 (...)
GRANT SELECT, INSERT ON T1 TO PUBLIC
END SQL
```

例 2) INPUTパラメータがあるSQL文

```
SQL
INSERT INTO SomeTable(Column1, Column2)
SELECT TRIM(A.Firstname) || ' ' || TRIM(A.Lastname), B.Whatever
FROM Table1 A, Table2 B
WHERE A.PkColumn = B.FkColumn
AND A.Filter = :arr[j].element
AND B.Filter MATCHES :str
END SQL
```

例 3) OUTPUTパラメータがあるSQL文

```
SQL
SELECT TRIM(TRAILING FROM Firstname) || ' ' || TRIM(LEADING FROM Lastname)
FROM Table1
WHERE PkColumn = 1
END SQL INTO str
```

例 4) INPUT及びOUTPUTパラメータ両方があるSQL文

```
SQL
SELECT SomeProcedure(ColName, :inval), OtherColumn
FROM dbase@server:SomeTable
WHERE PkColumn = :pkval
AND DtColumn > DATETIME(1970-01-01 00:00:00) YEAR TO SECOND
END SQL INTO x, y
```

尚、これらについての詳細は『Version7.3 4GL Reference Manuals』を参照ください。

3.6.3 フォーム機能の拡張

フォーム機能で DBCENTURY 機能を入力フィールドごとに違う設定で使用できるようになりました。

3.6.4 c4glスクリプトコマンドの変更

7.3x以前の IBM Informix4GL製品ではライブラリが静的にリンクされていましたが、7.3x4GL製品より、ライブラリを動的にリンクするように、c4glスクリプトコマンドが変更されました。従って、7.3x4GL製品で静的リンクを行う場合は“-static”オプションを指定して、コンパイルを行ってください。c4glスクリプトコマンドの変更によりUNIX環境変数でライブラリのパス指定が必要になります。UNIX環境変数でライブラリパスに、7.3Xの IBM Informix4GLがインストールされて

いる

ディレクトリにあるlib/toolsを指定してください。
それぞれのプラットフォームにより環境変数が異なりますので、詳細はOSベンダーにご確認ください。

以下は代表的なUNIXマシンの環境変数名です。

Sun	Solaris	LD_LIBRARY_PATH
HP	HP-UX	SHLIB_PATH
IBM	AIX	LIBPATH

例)

```
setenv LD_LIBRARY_PATH $INFORMIXDIR/lib/tools:$LD_LIBRARY_PATH
```

もし、ライブラリパスの指定の環境変数がない場合は、以下のファイルを/usr/libにコピーするか、もしくはリンクしてください。

`$INFORMIXDIR/lib/tools/lib4gsh730.so`

また、ご使用のプラットフォームが動的リンクをサポートしていない場合は、必ずオプション“-static”を使ってコンパイルを行ってください。

例) `c4gl -static -o custadd custadd.4gl`

4. プロジェクト作業編

この章では実際のマイグレーションに必要なタスクをステップ順に記述しています。

4.1 マイグレーションのための参考資料

IBM Informix Ver7.3Xtoolでは以下のマイグレーション資料を取り揃えています。
Ver7.3Xtoolへの移行作業にあたっては必ず詳細を参照してください。

『INFORMIX Migration Guide』	(日本語版あり)
『UNIX Products Migration Guide』	(日本語版あり)
『INFORMIX Installation Guide』	(日本語版あり)
『Guide to GLS Functionality』	(日本語版あり)

4.2 TERMCAPファイルの確認と修正

バージョン4.Xツールをお使いになっていたお客様は以下のtermcapファイルの内容を確認、及び修正の必要があります。

(1) 反転、グラフィック罫線を使用している場合

バージョン4.XツールとVer 7.3Xtoolではtermcapの解釈が一部異なっております。バージョン4.Xツールでは反転表示の移動に伴う初期化をTool内部で行なっていましたが、Ver7.3Xtoolでは反転表示の移動に伴う初期化については移動前にtermcapファイル内に初期化パラメータが必要です。初期化のパラメータにつきましては、ターミナル毎に異なる場合がありますので、ご使用のターミナルに併せて修正してください。

(2) ASIAN LANGUAGE SUPPORT ENHANCEMENTのサポートについて

IBM Informix が提供している\$INFORMIXDIR/etc/termcapをvt100でお使い頂くお客様については、インストール後の標準記述がASIAN LANGUAGE SUPPORT ENHANCMENT用に設定されておりません。
このため\$INFORMIXDIR/etc/termcapに記載しています2行を置き変えてください。

< 置き換え後の例 >

```
:gs=¥E(0;ge=¥E(B;gb=lmkjxq;¥
:ZG#0:ZA=¥E[0%?%p1%{0}%>%p1%{4}%<%&%t;1%;%?%p2%t;7%;%?%p3%t;5%;%? %p4%t;4%;m:
```

注意 1 : termcap ファイル中の記述で 2 行にわたる場合、改行されていると正常に動作しません。

注意 2 : この置き換え後の記述の例は基本サンプルです。お客様がお使い頂くターミナルソフトによりサポートされている機能が異なります。ターミナルソフトがサポートしています機能については、ターミナルソフトの提供元にお尋ね頂きお客様の使用環境に合わせてお客様自身でtermcapを調整していただく必要があります。

4.3 インストール

7.3Xtool製品については新しいディレクトリに新規インストールすることをお勧めします。もし、Client SDK 製品と組み合わせて同じディレクトリにインストールする場合、7.3Xtool製品のインストールが終了してから、必ず Client SDKをインストールしてください。

4.4 再コンパイル

Ver4.X、Ver6.X、Ver7.2XのInformixSQL/Informix4GLのアプリケーションから、Ver7.3Xへ移行する場合も、必ずVer7.3Xtoolの IBM INFORMIX製品上で再コンパイルを行ってください。

尚、提供していない関数を使用している場合、ほとんどの関数はUNIX上のライブラリ関数で代用することが可能です。

UNIXシステムで提供されている関数についての詳細は、UNIXシステムに添付されているマニュアルを参照してください。

4.5 環境変数の確認

7.3Xtoolでは原則としてInformix.rcファイルは提供されておりませんが、使用することは可能です。

但し、以前のバージョンと7.3XtoolではInformix.rcファイルの記述方法が異なりますので注意が必要です。(以下の例を参照)

例)

記述方法	Ver6.0X以前のツール	7.3Xのツール
DB_LOCALE=ja_JP.ujis	DB_LOCALEにujisを設定	無視
DB_LOCALE ja_jp.ujis	無視	DB_LOCALEにujisを設定

Ver7.3Xでは"="は無視します。

また、Ver6.0X製品ではja_JP.XXXX(JPが大文字)のみのサポートでしたが、Ver7.3X以降ではja_jp.XXXX(jpが小文字)もサポートしております。

Ver 7.3XtoolとVer7.2以降のエンジンを使用して日本語をハンドリングする場合は、ja_jp.XXXXを設定することをお勧めいたします。

Informix製品が環境変数をアクセスする際の、優先順位がありますが、ここでは、その優先順位をご紹介しますので、マイグレーションする際、及びアプリケーション運用に役立ててください。

<優先順位> (優先順位が高い順に 1 > 2 > 3 > 4 と記述されています)

1. 環境変数(シェル)で定義されている値
2. ユーザーのホームディレクトリ(~/.Informix)の専用の環境変数構成ファイルで定義された値
3. 共通の環境構成ファイル(\$INFORMIXDIR/etc/Informix.rc)で定義された値
4. デフォルトの値(いちばん低い優先順位)